BIKE SHARING DEMAND PREDICTION

High Level Design Document

MAY 3, 2023

PRATIK

Contents

1. Introduction

1.1 What is High-Level design document?

The main purpose of this HLD documentation is to feature the required details of the project and supply the outline of the machine learning model and also the written code. This additionally provides the careful description on however the complete project has been designed end-to-end.

1.2. Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

2. Description

2.1 Problem Perspective

The problem perspective is to address the issue of ensuring the availability and accessibility of rental bikes to the public at the right time. The goal is to reduce waiting times and provide a stable supply of rental bikes in urban cities.

2.2 Problem Statement

The problem statement is to predict the bike count required at each hour to ensure a stable supply of rental bikes. This prediction is crucial for managing the inventory of rental bikes effectively and meeting the demand of users.

2.3 Purposed Solution

The proposed solution aims to develop a predictive model that can forecast the bike count required at each hour. By analysing historical data and considering relevant factors such as season, weather, temperature, and time of day, the model can provide accurate predictions to support the management of rental bike inventory.

2.4 Solution Improvements

To improve the solution, we could consider incorporating additional factors such as special events or holidays that may affect the demand for rental bikes. Additionally, implementing a real-time data feed or integrating external data sources could enhance the accuracy of the predictions.

2.5 Technical Requirements

There are not any special hardware needs needed for running this application, the user should have an basic interactive device that has access to the web and should have the fundamental understanding of providing the input. And for the backend half the server should run all the package that's needed for the process the provided information and to show the results.

2.6 Data Requirements

The data required for the solution includes historical rental bike data containing columns such as instant (unique identifier), dteday (date), season, year,month, hour, holiday, weekday, workingday, weathersit (weather situation), temp (temperature), atemp(apparent temperature), hum (humidity), windspeed, casual (casual users), registered (registered users), and cnt (total count).

2.7 Tool Used

- Python 3.8 is used because the programming language and frame works like numpy,
- Pandas, sklearn and alternative modules for building the model.
- VSCODE can be used as IDE.
- For visualizations seaborn and components of matplotlib are being used.
- Front end development is completed by using Flask/Streamlit.
- GitHub is employed for version management and control.
- Finally Heroku/Streamlit is used for deployment.

2.8 Constraints

- No information was provided regarding special events which led to an increase or decrease in bike-sharing demand during these occasions.

- Small volatile dataset restricted the accuracy of our model prediction.
- To improve the solution, we could consider incorporating additional factors such as special events or holidays that may affect the demand for rental bikes.
- Additionally, implementing a real-time data feed or integrating external data sources could enhance the accuracy of the predictions.

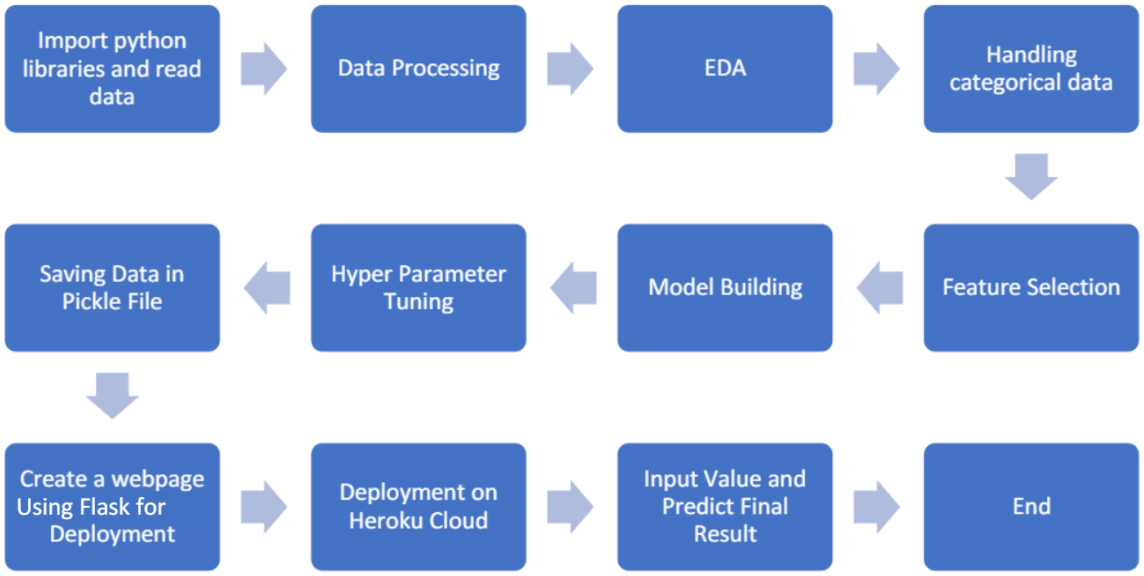
2.9 Assumptions

a) Historical bike usage patterns are indicative of future demand.
Explanation: It is assumed that the past patterns of bike rentals during specific hours, days, and months can provide insights into future demand and usage trends.

b) Hourly bike demand is influenced by weekday, weekend, and holiday factors.
Explanation: It is assumed that the demand for rental bikes differs based on weekdays, weekends, and holidays due to variations in people's routines, work schedules, and leisure activities. Due to variations in people's routines, work schedules, and leisure activities.

3 Design Flow

3.1 Modelling and Deployment Process



3.2 Logging

Each step is being logged within the system that runs internally, that shows the date time and therefore the processed that has been performed, work is completed in several layers as information, DEBUG, ERROR, WARNINGS. this provides the perceive of the logged info.

3.3 Exception Handling

Once ever a slip is occurred, the reason are logged in its several log file, in order that the developer will rectify the error.

4 Performance Evaluation

For performance evaluation, we will assess the accuracy and predictive capability of the developed model. We will utilize metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) to measure the goodness of fit between the predicted and actual bike counts. Additionally, we will employ cross-validation techniques to evaluate the model's robustness and generalization capabilities. The performance evaluation will also involve comparing the model against a baseline model or existing prediction methods used in the rental bike industry.

4.1 Reusability

The developed solution exhibits potential for reusability in various contexts. The predictive modelling framework and techniques employed can be adapted to other rental bike systems in different urban cities. The data preprocessing and feature engineering steps can serve as a template for similar prediction tasks involving temporal and weather-related data. The modularity and extensibility of the codebase enable the reuse of specific components, such as data transformation functions or feature selection algorithms, in other predictive modelling projects

4.2 Application Compatibility

The developed solution is designed to be compatible with a wide range of applications and platforms. It is implemented using widely-used programming languages such as Python, which ensures compatibility with popular data science libraries and frameworks. The model can be integrated into existing rental bike systems through Flask API interfaces or batch processing pipelines. The solution is platform-agnostic and can be deployed on cloud platforms, local servers, or embedded systems, providing flexibility and compatibility across various deployment environments.

4.3 Resource Utilization

The solution has been optimized to utilize computational resources efficiently. Through techniques like feature selection and dimensionality reduction, the model aims to reduce the computational burden while maintaining predictive performance. Additionally, resource monitoring and profiling can be employed during the model training and prediction phases to identify any potential bottlenecks or areas for further optimization. The goal is to ensure efficient resource utilization and minimize computational overhead, allowing the solution to scale and perform well even with large datasets or real-time prediction demands.

4.4 Deployment

The model is being deployed on Heroku/Streamlit

5 Conclusion

1) The model built using the LightGBM algorithm is the most accurate one.

2) Decision tree-based algorithms are more accurate than linear regression-based algorithms.

3) If not tuned properly, the Randomforest Regressor can overfit the training data, which can lead to poor generalization on the test data.

4) Hyperparameter tuning time is pretty high for the larger dataset in Randomforest and XG-Boost. Thus there is a trade-off between model accuracy and time consumed for model creation. So according to the desired requirement of the company model should be selected i.e if high accuracy is desired then LightGBM the best model else if one is dealing with a huge dataset and time is a constrain along with model interpretability is desired then the extra decision tree model can be selected.